Power Aware Wireless File Downloading: A Lyapunov Indexing Approach to A Constrained Restless Bandit Problem

Xiaohan Wei and Michael J. Neely

***Abstract*—This paper treats power-aware throughput maximization in a multi-user file downloading system. Each user can receive a new file only after its previous file is finished. The file state processes for each user act as coupled Markov chains that form a generalized restless bandit system. First, an optimal algorithm is derived for the case of one user. The algorithm maximizes throughput subject to an average power constraint. Next, the one-user algorithm is extended to a low complexity heuristic for the multi-user problem. The heuristic uses a simple online index policy. In a special case with no power-constraint, the multi-user heuristic is shown to be throughput optimal. Simulations are used to demonstrate effectiveness of the heuristic in the general case. For simple cases where the optimal solution can be computed offline, the heuristic is shown to be near-optimal for a wide range of parameters.**

I. Introduction

Consider a wireless access point, such as a base station or femto node, that delivers files to N different wireless users. The system operates in slotted time with time slots $t \in \{0, 1, 2, \ldots\}$. Each user can download at most one file at a time. File sizes are random and complete delivery of a file requires a random number of time slots. A new file request is made by each user at a random time after it finishes its previous download. Let $F_n(t) \in \{0, 1\}$ represent the binary *file state process* for user $n \in \{1, \ldots, N\}$. The state $F_n(t) = 1$ means that user n is currently active downloading a file, while the state $F_n(t) = 0$ means that user n is currently idle.

Idle times are assumed to be independent and geometrically distributed with parameter λ_n for each user n, so that the average idle time is $1/\lambda_n$. Active times depend on the random file size and the transmission decisions that are made. Every slot t, the access point observes which users are active and decides to serve a subset of at most M users, where M is the maximum number of simultaneous transmissions allowed in the system ($M < N$ is assumed throughout). The goal is to maximize a weighted sum of throughput subject to a total average power constraint.

The file state processes $F_n(t)$ are coupled controlled Markov chains that form a total state $(F_1(t), \ldots, F_N(t))$ that can be viewed as a *restless multi-armed bandit system*. Such problems are complex due to the inherent curse of dimensionality.

This paper first computes an online optimal algorithm for 1-user systems, i.e., the case $N = 1$. This simple case avoids the curse of dimensionality and provides valuable intuition. The optimal policy here is nontrivial and uses the theory of Lyapunov optimization for renewal systems [2][3]. The resulting algorithm makes a greedy transmission decision that affects success probability and power usage. The decision is based on a *drift-plus-penalty* index. Next, the algorithm is extended as a low complexity online heuristic for the N-user problem. The heuristic has the following desirable properties:

- Implementation of the N-user heuristic is as simple as comparing indices for N different 1-user problems.
- The N-user heuristic is analytically shown to meet the desired average power constraint.
- The N-user heuristic is shown in simulation to perform well over a wide range of parameters. Specifically, it is very close to optimal for example cases where an offline optimal can be computed.
- The N-user heuristic is shown to be optimal in a special case with no power constraint and with certain additional assumptions. The optimality proof uses a theory of *stochastic coupling* for queueing systems [4].

Prior work on wireless optimization uses Lyapunov functions to maximize throughput in cases where the users are assumed to have an infinite amount of data to send [4][5][6][7][8][9][10], or when data arrives according to a fixed rate process that does not depend on delays in the network (which necessitates *dropping* data if the arrival rate vector is outside of the capacity region) [6][8]. These models do not consider the interplay between arrivals at the transport layer and file delivery at the network layer. For example, a web user in a coffee shop may want to evaluate the file she downloaded before initiating another download. The current paper captures this interplay through the binary file state processes $F_n(t)$. This creates a complex problem of coupled Markov chains. This problem is fundamental to file downloading systems. The modeling and analysis of these systems is a significant contribution of the current paper.

To understand this issue, suppose the data arrival rate is fixed and does not adapt to the service received over the network. If this arrival rate exceeds network capacity by a factor of two, then at least half of all data must be dropped.

The authors are with the Electrical Engineering department at the University of Southern California, Los Angeles, CA.

This material was presented in part at the 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), Hammamet, Tunisia, 2014 [1].

This material is supported in part by one or more of: the NSF grants CCF-0747525, the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory W911NF-09-2-0053, The Okawa Foundation research award.

This can result in an unusable data stream, possibly one that contains every odd-numbered packet. A more practical model assumes that full files must be downloaded and that new downloads are only initiated when previous ones are completed. A general model in this direction would allow each user to download up to K files simultaneously. This paper considers the case $K = 1$, so that each user is either actively downloading a file, or is idle.[1] The resulting system for N users has a nontrivial Markov structure with 2^N states.

Markov decision problems (MDPs) can be solved offline via linear programming [11]. This can be prohibitively complex for large dimensional problems. Low complexity solutions for coupled MDPs are possible in special cases when the coupling involves only time average constraints [12]. Finite horizon coupled MDPs are treated via integer programming in [13] and via a heuristic "task decomposition" method in [14]. The problem of the current paper does not fit the framework of [12]-[14] because it includes both time-average constraints (on average power expenditure) and instantaneous constraints which restrict the number of users that can be served on one slot. The latter service restriction is similar to a traditional restless multi-armed bandit (RMAB) system [15].

RMAB problems consider a population of N parallel MDPs that continue evolving whether in operation or not (although under different rules). The goal is to choose the MDPs in operation during each time slot so as to maximize the expected reward subject to a constraint on the number of MDPs in operation. The problem is in general complex (see P-SPACE hardness results in [16]). A standard low-complexity heuristic for such problems is the *Whittle's index* technique [15]. **However, the Whittle's index framework applies only when there are two options on each state (active and passive). Further, it does not consider the additional time average cost constraints. The *Lyapunov indexing* algorithm developed in the current paper can be viewed as an alternative indexing scheme that can always be implemented and that incorporates additional time average constraints.** It is likely that the techniques of the current paper can be extended to other constrained RMAB problems. Prior work in [4] develops a Lyapunov drift method for queue stability, and work in [2] develops a drift-plus-penalty ratio method for optimization over renewal systems. The current work is the first to use these techniques as a low complexity heuristic for multidimensional Markov problems.

Work in [4] uses the theory of stochastic coupling to show that a *longest connected queue* algorithm is delay optimal in a multi-dimensional queueing system with *special symmetric assumptions*. The problem in [4] is different from that of the current paper. However, a similar coupling approach is used in Section IV to show that, for a special case with no power constraint, the Lyapunov indexing algorithm is throughput optimal in certain *asymmetric* cases. As a consequence, the proof shows the policy is also optimal for a different setting with M servers, N single-buffer queues, and arbitrary packet arrival rates $(\lambda_1, \ldots, \lambda_N)$.

[1]One way to allow a user n to download up to K files simultaneously is as follows: Define K *virtual users* with separate binary file state processes. The transition probability from idle to active in each of these virtual users is λ_n/K. The conditional rate of total new arrivals for user n (given that m files are currently in progress) is then $\lambda_n(1-m/K)$ for $m \in \{0, 1, \ldots, M\}$.

II. Single user scenario

Consider a file downloading system that consists of only one user that repeatedly downloads files. Let $F(t) \in \{0, 1\}$ be the file state process of the user. State "1" means there is a file in the system that has not completed its download, and "0" means no file is waiting. The length of each file is independent and is either exponentially distributed or geometrically distributed (described in more detail below). Let $\overline{B}$ denote the expected file size in bits. Time is slotted. At each slot in which there is an active file for downloading, the user makes a service decision that affects both the downloading success probability and the power expenditure. After a file is downloaded, the system goes idle (state 0) and remains in the idle state for a random amount of time that is independent and geometrically distributed with parameter $\lambda > 0$.

A transmission decision is made on each slot t in which $F(t) = 1$. The decision affects the number of bits that are sent, the probability these bits are successfully received, and the power usage. Let $\alpha(t)$ denote the decision variable at slot t and let $\mathcal{A}$ represent an abstract action set with a finite number of elements. The set $\mathcal{A}$ can represent a collection of modulation and coding options for each transmission. Assume also that $\mathcal{A}$ contains an idle action denoted as "0." The decision $\alpha(t)$ determines the following two values:

- The probability of successfully downloading a file $\phi(\alpha(t))$, where $\phi(\cdot) \in [0, 1]$ with $\phi(0) = 0$.
- The power expenditure $p(\alpha(t))$, where $p(\cdot)$ is a nonnegative function with $p(0) = 0$.

The user chooses $\alpha(t) = 0$ whenever $F(t) = 0$. The user chooses $\alpha(t) \in \mathcal{A}$ for each slot t in which $F(t) = 1$, with the goal of maximizing throughput subject to a time average power constraint.

The problem can be described by a two state Markov decision process with binary state $F(t)$. Given $F(t) = 1$, a file is currently in the system. This file will finish its download at the end of the slot with probability $\phi(\alpha(t))$. Hence, the transition probabilities out of state 1 are:

$$Pr[F(t+1) = 0 | F(t) = 1] = \phi(\alpha(t)) \quad (1)$$

$$Pr[F(t+1) = 1 | F(t) = 1] = 1 - \phi(\alpha(t)) \quad (2)$$

Given $F(t) = 0$, the system is idle and will transition to the active state in the next slot with probability λ, so that:

$$Pr[F(t+1) = 1 | F(t) = 0] = \lambda \quad (3)$$

$$Pr[F(t+1) = 0 | F(t) = 0] = 1 - \lambda \quad (4)$$

Define the throughput, measured by bits per slot, as:

$$\liminf_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \overline{B}\phi(\alpha(t))$$

The file downloading problem reduces to the following:

$$\text{Maximize: } \liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \overline{B}\phi(\alpha(t)) \tag{5}$$

$$\text{Subject to: } \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} p(\alpha(t)) \leq \beta \tag{6}$$

$$\alpha(t) \in \mathcal{A}\ \forall t \in \{0,1,2,\ldots\} \text{ such that } F(t)=1 \tag{7}$$

$$\text{Transition probabilities satisfy (1)-(4)} \tag{8}$$

where β is a positive constant that determines the desired average power constraint.

A. The memoryless file size assumption

The above model assumes that file completion success on slot t depends only on the transmission decision $\alpha(t)$, independent of history. This implicitly assumes that file length distributions have a *memoryless property* where the residual file length is independent of the amount already delivered. Further, it is assumed that if the controller selects a transmission rate that is larger than the residual bits in the file, the remaining portion of the transmission is padded with *fill bits*. This ensures error events provide no information about the residual file length beyond the already known 0/1 binary file state. Of course, error probability might be improved by removing padded bits. However, this affects only the last transmission of a file and has negligible impact when expected file size is large in comparison to the amount that can be transmitted in one slot. Note that padding is not needed in the special case when all transmissions send one fixed length packet.

The memoryless property holds when each file i has independent length B_i that is *exponentially distributed* with mean length $\overline{B}$ bits, so that:

$$Pr[B_i > x] = e^{-x/\overline{B}} \text{ for } x > 0$$

For example, suppose the *transmission rate* $r(t)$ (in units of bits/slot) and the *transmission success probability* $q(t)$ are given by general functions of $\alpha(t)$:

$$\begin{aligned} r(t) &= \hat{r}(\alpha(t)) \\ q(t) &= \hat{q}(\alpha(t)) \end{aligned}$$

Then the file completion probability $\phi(\alpha(t))$ is the probability that the *residual* amount of bits in the file is less than or equal to $r(t)$, *and* that the transmission of these residual bits is a success. By the memoryless property of the exponential distribution, the residual file length is distributed the same as the original file length. Thus:

$$\begin{aligned} \phi(\alpha(t)) &= \hat{q}(\alpha(t))Pr[B_i \leq \hat{r}(\alpha(t))] \\ &= \hat{q}(\alpha(t))\int_0^{\hat{r}(\alpha(t))} \frac{1}{\overline{B}}e^{-x/\overline{B}}dx \end{aligned} \tag{9}$$

Alternatively, history independence holds when each file i consists of a random number Z_i of fixed length packets, where Z_i is geometrically distributed with mean $\overline{Z} = 1/\mu$. Assume each transmission sends exactly one packet, but different power levels affect the transmission success probability $q(t) = \hat{q}(\alpha(t))$. Then:

$$\phi(\alpha(t)) = \mu\hat{q}(\alpha(t)) \tag{10}$$

The memoryless file length assumption allows the file state to be modeled by a simple binary-valued process $F(t) \in \{0,1\}$. However, actual file sizes may not have an exponential or geometric distribution. One way to treat general distributions is to *approximate* the file sizes as being memoryless by using a $\phi(\alpha(t))$ function defined by either (9) or (10), formed by matching the average file size $\overline{B}$ or average number of packets $\overline{Z}$. The *decisions* $\alpha(t)$ are made according to the algorithm below, but the actual event outcomes that arise from these decisions are not memoryless. A simulation comparison of this approximation is provided in Section V, where it is shown to be remarkably accurate (see Fig. 7).

The algorithm in this section optimizes over the class of all algorithms that do not use residual file length information. This maintains low complexity by ensuring a user has a binary-valued Markov state $F(t) \in \{0,1\}$. While a system controller might know the residual file length, incorporating this knowledge creates a Markov decision problem with an infinite number of states (one for each possible value of residual length) which significantly complicates the scenario.

B. Lyapunov optimization

This subsection develops an online algorithm for problem (5)-(8). First, notice that file state "1" is recurrent under any decisions for $\alpha(t)$. Denote t_k as the k-th time when the system returns to state "1." Define the *renewal frame* as the time period between t_k and t_{k+1}. Define the *frame size*:

$$T[k] = t_{k+1} - t_k$$

Notice that $T[k] = 1$ for any frame k in which the file does not complete its download. If the file is completed on frame k, then $T[k] = 1 + G_k$, where G_k is a geometric random variable with mean $\mathbb{E}[G_k] = 1/\lambda$. Each frame k involves only a single decision $\alpha(t_k)$ that is made at the beginning of the frame. Thus, the total power used over the duration of frame k is:

$$\sum_{t=t_k}^{t_{k+1}-1} p(\alpha(t)) = p(\alpha(t_k)) \tag{11}$$

Using a technique similar to that proposed in [2], we treat the time average constraint in (6) using a virtual queue $Q[k]$ that is updated every frame k by:

$$Q[k+1] = \max\{Q[k] + p(\alpha(t_k)) - \beta T[k],\ 0\} \tag{12}$$

with initial condition $Q[0] = 0$. The algorithm is then parameterized by a constant $V \geq 0$ which affects a performance tradeoff. At the beginning of the k-th renewal frame, the user observes virtual queue $Q[k]$ and chooses $\alpha(t_k)$ to maximize the following drift-plus-penalty (DPP) ratio [2]:

$$\max_{\alpha(t_k)\in\mathcal{A}} \frac{V\overline{B}\phi(\alpha(t_k)) - Q[k]p(\alpha(t_k))}{\mathbb{E}[T[k]|\alpha(t_k)]} \tag{13}$$

The numerator of the above ratio adds a "queue drift term" $-Q[k]p(\alpha(t_k))$ to the "current reward term" $V\overline{B}\phi(\alpha(t_k))$. The intuition is that it is desirable to have a large value of current reward, but it is also desirable to have a large drift (since this tends to decrease queue size). Creating a weighted sum of these two terms and dividing by the expected frame size gives a simple index. The next subsections show that, for the context of the current paper, this index leads to an algorithm that pushes throughput arbitrarily close to optimal (depending on the chosen V parameter) with a strong sample path guarantee on average power expenditure.

The denominator in (13) can easily be computed:

$$\mathbb{E}[T[k]|\alpha(t_k)] = 1 + \frac{\phi(\alpha(t_k))}{\lambda}$$

Thus, (13) is equivalent to

$$\max_{\alpha(t_k)\in\mathcal{A}} \frac{V\overline{B}\phi(\alpha(t_k)) - Q[k]p(\alpha(t_k))}{1+\phi(\alpha(t_k))/\lambda} \tag{14}$$

Since there are only a finite number of elements in $\mathcal{A}$, (14) is easily computed. This gives the following algorithm for the single-user case:

Algorithm 1:

- At each time t_k, the user observes virtual queue $Q[k]$ and chooses $\alpha(t_k)$ as the solution to (14) (where ties are broken arbitrarily).
- The value $Q[k+1]$ is computed according to (12) at the end of the k-th frame.

C. Average power constraints via queue bounds

Lemma 1: If there is a constant $C \geq 0$ such that $Q[k] \leq C$ for all $k \in \{0, 1, 2, \ldots\}$, then:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} p(\alpha(t)) \leq \beta$$

Proof: From (12), we know that for each frame k:

$$Q[k+1] \geq Q[k] + p(\alpha(t_k)) - T[k]\beta$$

Rearranging terms and using $T[k] = t_{k+1} - t_k$ gives:

$$p(\alpha(t_k)) \leq (t_{k+1} - t_k)\beta + Q[k+1] - Q[k]$$

Fix $K > 0$. Summing over $k \in \{0, 1, \cdots, K-1\}$ gives:

$$\begin{aligned}\sum_{k=0}^{K-1} p(\alpha(t_k)) &\leq (t_K - t_0)\beta + Q[K] - Q[0] \\ &\leq t_K\beta + C\end{aligned}$$

The sum power over the first K frames is the same as the sum up to time $t_K - 1$, and so:

$$\sum_{t=0}^{t_K-1} p(\alpha(t)) \leq t_K\beta + C$$

Dividing by t_K gives:

$$\frac{1}{t_K}\sum_{t=0}^{t_K-1} p(\alpha(t)) \leq \beta + C/t_K.$$

Taking $K \to \infty$, then,

$$\limsup_{K\to\infty} \frac{1}{t_K}\sum_{t=0}^{t_K-1} p(\alpha(t)) \leq \beta \tag{15}$$

Now for each positive integer T, let $K(T)$ be the integer such that $t_{K(T)} \leq T < t_{K(T)+1}$. Since power is only used at the first slot of a frame, one has:

$$\frac{1}{T}\sum_{t=0}^{T-1} p(\alpha(t)) \leq \frac{1}{t_{K(T)}}\sum_{t=0}^{t_{K(T)}-1} p(\alpha(t))$$

Taking a $\limsup$ as $T \to \infty$ and using (15) yields the result. ■

The next lemma shows that the queue process under our proposed algorithm is deterministically bounded. Define:

$$\begin{aligned} p^{min} &= \min_{\alpha\in\mathcal{A}\setminus\{0\}} p(\alpha) \\ p^{max} &= \max_{\alpha\in\mathcal{A}\setminus\{0\}} p(\alpha)\end{aligned}$$

Assume that $p^{min} > 0$.

Lemma 2: If $Q[0] = 0$, then under our algorithm we have for all $k > 0$:

$$Q[k] \leq \max\left\{\frac{V\overline{B}}{p^{min}} + p^{max} - \beta, 0\right\}$$

Proof: First, consider the case when $p^{max} \leq \beta$. From (12) and the fact that $T[k] \geq 1$ for all k, it is clear the queue can never increase, and so $Q[k] \leq Q[0] = 0$ for all $k > 0$.

Next, consider the case when $p^{max} > \beta$. We prove the assertion by induction on k. The result trivially holds for $k = 0$. Suppose it holds at $k = l$ for $l > 0$, so that:

$$Q[l] \leq \frac{V\overline{B}}{p^{min}} + p^{max} - \beta$$

We are going to prove that the same holds for $k = l+1$. There are two cases:

1) $Q[l] \leq \frac{V\overline{B}}{p^{min}}$. In this case we have by (12):

$$\begin{aligned} Q[l+1] &\leq Q[l] + p^{max} - \beta \\ &\leq \frac{V\overline{B}}{p^{min}} + p^{max} - \beta\end{aligned}$$

2) $\frac{V\overline{B}}{p^{min}} < Q[l] \leq \frac{V\overline{B}}{p^{min}} + p^{max} - \beta$. In this case, if $p(\alpha(t_l)) = 0$ then the queue cannot increase, so:

$$Q[l+1] \leq Q[l] \leq \frac{V\overline{B}}{p^{min}} + p^{max} - \beta$$

On the other hand, if $p(\alpha(t_l)) > 0$ then $p(\alpha(t_l)) \geq p^{min}$ and so the numerator in (14) satisfies:

$$\begin{aligned} V\overline{B}\phi(\alpha(t_l)) - Q[l]p(\alpha(t_l)) &\leq V\overline{B} - Q[l]p^{min} \\ &< 0\end{aligned}$$

and so the maximizing ratio in (14) is negative. However, the maximizing ratio in (14) *cannot* be negative because the alternative choice $\alpha(t_l) = 0$ increases the ratio to 0. This contradiction implies that we cannot have $p(\alpha(t_l)) > 0$.

■

The above is a *sample path result* that only assumes parameters satisfy $\lambda > 0$, $\overline{B} > 0$, and $0 \leq \phi(\cdot) \leq 1$. Thus, the algorithm meets the average power constraint even if it uses incorrect values for these parameters. The next subsection provides a throughput optimality result when these parameters match the true system values.

D. Optimality over randomized algorithms

Consider the following class of *i.i.d. randomized algorithms*: Let $\theta(\alpha)$ be non-negative numbers defined for each $\alpha \in \mathcal{A}$, and suppose they satisfy $\sum_{\alpha\in\mathcal{A}} \theta(\alpha) = 1$. Let $\alpha^*(t)$ represent a policy that, every slot t for which $F(t) = 1$, chooses $\alpha^*(t) \in \mathcal{A}$ by independently selecting strategy α with probability $\theta(\alpha)$. Then $(p(\alpha^*(t_k)), \phi(\alpha^*(t_k)))$ are independent and identically distributed (i.i.d.) over frames k. Under this algorithm, it follows by the law of large numbers that the throughput and power expenditure satisfy (with probability 1):

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\overline{B}\phi(\alpha^*(t)) &= \frac{\overline{B}\mathbb{E}\left[\phi(\alpha^*(t_k))\right]}{1+\mathbb{E}\left[\phi(\alpha^*(t_k))\right]/\lambda}\\ \lim_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}p(\alpha^*(t)) &= \frac{\mathbb{E}\left[p(\alpha^*(t_k))\right]}{1+\mathbb{E}\left[\phi(\alpha^*(t_k))\right]/\lambda}\end{aligned}$$

It can be shown that optimality of problem (5)-(8) can be achieved over this class [3]. Thus, there exists an i.i.d. randomized algorithm $\alpha^*(t)$ that satisfies:

$$\frac{\overline{B}\mathbb{E}\left[\phi(\alpha^*(t_k))\right]}{1+\mathbb{E}\left[\phi(\alpha^*(t_k))\right]/\lambda} = \mu^* \quad (16)$$

$$\frac{\mathbb{E}\left[p(\alpha^*(t_k))\right]}{1+\mathbb{E}\left[\phi(\alpha^*(t_k))\right]/\lambda} \leq \beta \quad (17)$$

where μ^* is the optimal throughput for the problem (5)-(8).

E. Key feature of the drift-plus-penalty ratio

Define $\mathcal{H}[k]$ as the *system history* up to frame k, which includes the actions taken $\alpha[0], \cdots, \alpha[k-1]$ frame lengths $T[0], \cdots, T[k-1]$, the busy period in each frame, the idle period in each frame, and the queue value $Q[k]$ (since this is determined by the random events before frame k). Consider the algorithm that, on frame k, observes $Q[k]$ and chooses $\alpha(t_k)$ according to (14). The following key feature of this algorithm can be shown (see [2] for related results):

$$\begin{aligned}&\frac{\mathbb{E}\left[-V\overline{B}\phi(\alpha(t_k)) + Q[k]p(\alpha(t_k))|\mathcal{H}[k]\right]}{\mathbb{E}\left[1+\phi(\alpha(t_k))/\lambda|\mathcal{H}[k]\right]}\\ &\leq \frac{\mathbb{E}\left[-V\overline{B}\phi(\alpha^*(t_k)) + Q[k]p(\alpha^*(t_k))|\mathcal{H}[k]\right]}{\mathbb{E}\left[1+\phi(\alpha^*(t_k))/\lambda|\mathcal{H}[k]\right]}\end{aligned}$$

where $\alpha^*(t_k)$ is any (possibly randomized) alternative decision that is based only on $\mathcal{H}[k]$. This is an intuitive property: By design, the algorithm in (14) observes $\mathcal{H}[k]$ and then chooses a particular action $\alpha(t_k)$ to minimize the ratio over all deterministic actions. Thus, as can be shown, it also minimizes the ratio over all potentially randomized actions. Using the (randomized) i.i.d. decision $\alpha^*(t_k)$ from (16)-(17) in the above and noting that this alternative decision is independent of $\mathcal{H}[k]$ gives:

$$\frac{\mathbb{E}\left[-V\overline{B}\phi(\alpha(t_k)) + Q[k]p(\alpha(t_k))|\mathcal{H}[k]\right]}{\mathbb{E}\left[1+\phi(\alpha(t_k))/\lambda|\mathcal{H}[k]\right]} \leq -V\mu^* + Q[k]\beta \quad (18)$$

F. Performance theorem

Theorem 1: The proposed algorithm achieves the constraint $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}p(\alpha(t)) \leq \beta$ and yields throughput satisfying (with probability 1):

$$\liminf_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\overline{B}\phi(\alpha(t)) \geq \mu^* - \frac{C_0}{V} \quad (19)$$

where C_0 is a constant.[2]

Proof: First, for any fixed V, Lemma 2 implies that the queue is deterministically bounded. Thus, according to Lemma 1, the proposed algorithm achieves the constraint $\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}p(\alpha(t)) \leq \beta$. The rest is devoted to proving the throughput guarantee (19).

Define:

$$L(Q[k]) = \frac{1}{2}Q[k]^2.$$

We call this a *Lyapunov function*. Define a frame-based Lyapunov Drift as:

$$\Delta[k] = L(Q[k+1]) - L(Q[k])$$

According to (12) we get

$$Q[k+1]^2 \leq \left(Q[k] + p(\alpha(t_k)) - T[k]\beta\right)^2.$$

Thus:

$$\Delta[k] \leq \frac{(p(\alpha(t_k)) - T[k]\beta)^2}{2} + Q[k](p(\alpha(t_k)) - T[k]\beta)$$

Taking a conditional expectation of the above given $\mathcal{H}[k]$ and recalling that $\mathcal{H}[k]$ includes the information $Q[k]$ gives:

$$\mathbb{E}\left[\Delta[k]|\mathcal{H}[k]\right] \leq C_0 + Q[k]\mathbb{E}\left[p(\alpha(t_k)) - \beta T[k]|\mathcal{H}[k]\right] \quad (20)$$

where C_0 is a constant that satisfies the following for all possible histories $\mathcal{H}[k]$:

$$\mathbb{E}\left[\frac{(p(\alpha(t_k)) - T[k]\beta)^2}{2} \mid \mathcal{H}[k]\right] \leq C_0$$

Such a constant C_0 exists because the power $p(\alpha(t_k))$ is deterministically bounded, and the frame sizes $T[k]$ are bounded in second moment regardless of history.

Adding the "penalty" $-\mathbb{E}\left[V\overline{B}\phi(\alpha(t_k))|\mathcal{H}[k]\right]$ to both sides of (20) gives:

$$\begin{aligned}&\mathbb{E}\left[\Delta[k] - V\overline{B}\phi(\alpha(t_k))|\mathcal{H}[k]\right]\\ &\leq C_0 + \mathbb{E}\left[-V\overline{B}\phi(\alpha(t_k)) + Q[k](p(\alpha(t_k)) - \beta T[k])|\mathcal{H}[k]\right]\\ &= C_0 - Q[k]\beta\mathbb{E}\left[T[k]|\mathcal{H}[k]\right]\\ &+ \frac{\mathbb{E}\left[T[k]|\mathcal{H}[k]\right]\mathbb{E}\left[-V\overline{B}\phi(\alpha(t_k)) + Q[k]p(\alpha(t_k))|\mathcal{H}[k]\right]}{\mathbb{E}\left[T[k]|\mathcal{H}[k]\right]}\end{aligned}$$

[2]The constant C_0 is independent of V and is given in the proof.

Expanding $T[k]$ in the denominator of the last term gives:

$$\begin{aligned}&\mathbb{E}\left[\Delta[k] - V\overline{B}\phi(\alpha(t_k))|\mathcal{H}[k]\right] \\ &\leq C_0 - Q[k]\beta\mathbb{E}\left[T[k]|\mathcal{H}[k]\right] + \mathbb{E}\left[T[k]|\mathcal{H}[k]\right] \times \\ &\frac{\mathbb{E}\left[-V\overline{B}\phi(\alpha(t_k)) + Q[k]p(\alpha(t_k))|\mathcal{H}[k]\right]}{\mathbb{E}\left[1 + \phi(\alpha(t_k))/\lambda|\mathcal{H}[k]\right]}\end{aligned}$$

Substituting (18) into the above expression gives:

$$\begin{aligned}&\mathbb{E}\left[\Delta[k] - V\overline{B}\phi(\alpha(t_k))|\mathcal{H}[k]\right] \\ &\leq C_0 - Q[k]\beta\mathbb{E}\left[T[k]|\mathcal{H}[k]\right] \\ &\quad +\mathbb{E}\left[T[k]|\mathcal{H}[k]\right](-V\mu^* + \beta Q[k]) \\ &= C_0 - V\mu^*\mathbb{E}\left[T[k]|\mathcal{H}[k]\right] \qquad (21)\end{aligned}$$

Rearranging gives:

$$\mathbb{E}\left[\Delta[k] + V(\mu^* T[k] - \overline{B}\phi(\alpha(t_k)))|\mathcal{H}[k]\right] \leq C_0 \qquad (22)$$

The above is a drift-plus-penalty expression. Because we already know the queue $Q[k]$ is deterministically bounded, it follows that:

$$\sum_{k=1}^{\infty} \frac{\mathbb{E}\left[\Delta[k]^2\right]}{k^2} < \infty$$

This, together with (22), implies by the drift-plus-penalty result in Proposition 2 of [17] that (with probability 1):

$$\limsup_{K\to\infty} \frac{1}{K}\sum_{k=0}^{K-1}\left[\mu^* T[k] - \overline{B}\phi(\alpha(t_k))\right] \leq \frac{C_0}{V}$$

Thus, for any $\epsilon > 0$ one has for all sufficiently large K:

$$\frac{1}{K}\sum_{k=0}^{K-1}\left[\mu^* T[k] - \overline{B}\phi(\alpha(t_k))\right] \leq \frac{C_0}{V} + \epsilon$$

Rearranging implies that for all sufficiently large K:

$$\begin{aligned}\frac{\sum_{k=0}^{K-1}\overline{B}\phi(\alpha(t_k))}{\sum_{k=0}^{K-1}T[k]} &\geq \mu^* - \frac{(C_0/V + \epsilon)}{\frac{1}{K}\sum_{k=0}^{K-1}T[k]} \\ &\geq \mu^* - (C_0/V + \epsilon)\end{aligned}$$

where the final inequality holds because $T[k] \geq 1$ for all k. Thus:

$$\liminf_{K\to\infty} \frac{\sum_{k=0}^{K-1}\overline{B}\phi(\alpha(t_k))}{\sum_{k=0}^{K-1}T[k]} \geq \mu^* - (C_0/V + \epsilon)$$

The above holds for all $\epsilon > 0$. Taking a limit as $\epsilon \to 0$ implies:

$$\liminf_{K\to\infty} \frac{\sum_{k=0}^{K-1}\overline{B}\phi(\alpha(t_k))}{\sum_{k=0}^{K-1}T[k]} \geq \mu^* - C_0/V.$$

Notice that $\phi(\alpha(t))$ only changes at the boundary of each frame and remains 0 within the frame. Thus, we can replace the sum over frames k by a sum over slots t. The desired result follows. ■

The theorem shows that throughput can be pushed within $O(1/V)$ of the optimal value μ^*, where V can be chosen as large as desired to ensure throughput is arbitrarily close to optimal. The tradeoff is a queue bound that grows linearly with V according to Lemma 2, which affects the convergence time required for the constraints to be close to the desired time averages (as described in the proof of Lemma 1).

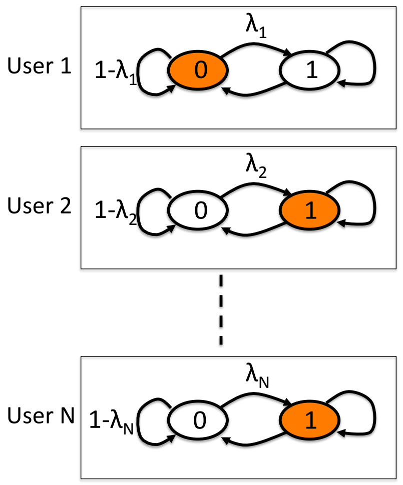

Fig. 1. A system with N users. The shaded node for each user n indicates the current file state $F_n(t)$ of that user. There are 2^N different state vectors.

III. Multi-user file downloading

This section considers a multi-user file downloading system that consists of N single-user subsystems. Each subsystem is similar to the single-user system described in the previous section. Specifically, for the n-th user (where $n \in \{1, \ldots, N\}$):

- The file state process is $F_n(t) \in \{0, 1\}$.
- The transmission decision is $\alpha_n(t) \in \mathcal{A}_n$, where $\mathcal{A}_n$ is an abstract set of transmission options for user n.
- The power expenditure on slot t is $p_n(\alpha_n(t))$.
- The success probability on a slot t for which $F_n(t) = 1$ is $\phi_n(\alpha_n(t))$, where $\phi_n(\cdot)$ is the function that describes file completion probability for user n.
- The idle period parameter is $\lambda_n > 0$.
- The average file size is $\overline{B}_n$ bits.

Assume that the random variables associated with different subsystems are mutually independent. The resulting Markov decision problem has 2^N states, as shown in Fig. 1. The transition probabilities for each active user depends on which users are selected for transmission and on the corresponding transmission modes. This is a *restless bandit system* because there can also be transitions for non-selected users (specifically, it is possible to transition from inactive to active).

To control the downloading process, there is a central server with only M threads ($M < N$), meaning that *at most M jobs can be processed simultaneously*. So at each time slot, the server has to make decisions selecting at most M out of N users to transmit a portion of their files. These decisions are further restricted by a global time average power constraint. The goal is to maximize the aggregate throughput, which is defined as

$$\liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N} c_n\overline{B}_n\phi(\alpha_n(t))$$

where $c_1, c_2, \ldots, c_N$ are a collection of positive weights that can be used to prioritize users. Thus, this multi-user file downloading problem reduces to the following:

$$\text{Max: } \liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N} c_n\overline{B}_n\phi_n(\alpha_n(t)) \tag{23}$$

$$\text{S.t.: } \limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N} p_n(\alpha_n(t)) \leq \beta \tag{24}$$

$$\sum_{n=1}^{N} I(\alpha_n(t)) \leq M \quad \forall t \in \{0,1,2,\cdots\} \tag{25}$$

$$Pr[F_n(t+1)=1 \mid F_n(t)=0] = \lambda_n \tag{26}$$

$$Pr[F_n(t+1)=0 \mid F_n(t)=1] = \phi_n(\alpha_n(t)) \tag{27}$$

where the constraints (26)-(27) hold for all $n \in \{1,\ldots,N\}$ and $t \in \{0,1,2,\ldots\}$, and where $I(\cdot)$ is the indicator function defined as:

$$I(x) = \begin{cases} 0, & \text{if } x = 0; \\ 1, & \text{otherwise.} \end{cases}$$

A. Lyapunov indexing algorithm

This section develops our indexing algorithm for the multi-user case using the single-user case as a stepping stone. The major difficulty is the instantaneous constraint $\sum_{n=1}^{N} I(\alpha_n(t)) \leq M$. Temporarily neglecting this constraint, we use Lyapunov optimization to deal with the time average power constraint first.

We introduce a virtual queue $Q(t)$, which is again 0 at $t = 0$. Instead of updating it on a frame basis, the server updates this queue every slot as follows:

$$Q(t+1) = \max\left\{Q(t) + \sum_{n=1}^{N} p_n(\alpha_n(t)) - \beta, 0\right\}. \tag{28}$$

Define $\mathcal{N}(t)$ as the set of users beginning their renewal frames at time t, so that $F_n(t) = 1$ for all such users. In general, $\mathcal{N}(t)$ is a subset of $\mathcal{N} = \{1,2,\cdots,N\}$. Define $|\mathcal{N}(t)|$ as the number of users in the set $\mathcal{N}(t)$.

At each time slot t, the server observes the queue state $Q(t)$ and chooses $(\alpha_1(t),\ldots,\alpha_N(t))$ in a manner similar to the single-user case. Specifically, for each user $n \in \mathcal{N}(t)$ define:

$$g_n(\alpha_n(t)) \triangleq \frac{Vc_n\overline{B}_n\phi_n(\alpha_n(t)) - Q(t)p_n(\alpha_n(t))}{1+\phi_n(\alpha_n(t))/\lambda_n} \tag{29}$$

This is similar to the expression (14) used in the single-user optimization. Call $g_n(\alpha_n(t))$ a *reward*. Now define an index for each subsystem n by:

$$\gamma_n(t) \triangleq \max_{\alpha_n(t)\in\mathcal{A}_n} g_n(\alpha_n(t)) \tag{30}$$

which is the maximum possible reward one can get from the n-th subsystem at time slot t. Thus, it is natural to define the following myopic algorithm: Find the (at most) M subsystems in $\mathcal{N}(t)$ with the greatest rewards, and serve these with their corresponding optimal $\alpha_n(t)$ options in $\mathcal{A}_n$ that maximize $g_n(\alpha_n(t))$.

Algorithm 2:

- At each time slot t, the server observes virtual queue state $Q(t)$ and computes the indices using (30) for all $n \in \mathcal{N}(t)$.
- Activate the $\min[M, |\mathcal{N}(t)|]$ subsystems with greatest indices, using their corresponding actions $\alpha_n(t) \in \mathcal{A}_n$ that maximize $g_n(\alpha_n(t))$.
- Update $Q(t)$ according to (28) at the end of each slot t.

B. Theoretical performance analysis

In this subsection, we show that the above algorithm always satisfies the desired time average power constraint. Define:

$$\begin{aligned} p_n^{min} &= \min_{\alpha_n\in\mathcal{A}_n\backslash\{0\}} p_n(\alpha_n) \\ p^{min} &= \min_n p_n^{min} \\ p_n^{max} &= \max_{\alpha_n\in\mathcal{A}_n} p_n(\alpha_n) \\ c^{max} &= \max_n c_n \\ \overline{B}^{max} &= \max_n \overline{B}_n \end{aligned}$$

Assume that $p^{min} > 0$.

Lemma 3: Under the above Lyapunov indexing algorithm, the queue $\{Q(t)\}_{t=0}^{\infty}$ is deterministically bounded. Specifically, we have for all $t \in \{0,1,2,\ldots\}$:

$$Q(t) \leq \max\left\{\frac{Vc^{max}\overline{B}^{max}}{p^{min}} + \sum_{n=1}^{N} p_n^{max} - \beta, 0\right\}$$

Proof: First, consider the case when $\sum_{n=1}^{N} p_n^{max} \leq \beta$. Since $Q(0) = 0$, it is clear from the updating rule (28) that $Q(t)$ will remain 0 for all t.

Next, consider the case when $\sum_{n=1}^{N} p_n^{max} > \beta$. We prove the assertion by induction on t. The result trivially holds for $t = 0$. Suppose at $t = t'$, we have:

$$Q(t') \leq \frac{Vc^{max}\overline{B}^{max}}{p^{min}} + \sum_{n=1}^{N} p_n^{max} - \beta$$

We are going to prove that the same statement holds for $t = t'+1$. We further divide it into two cases:

1) $Q(t') \leq \frac{Vc^{max}\overline{B}^{max}}{p^{min}}$. In this case, since the queue increases by at most $\sum_{n=1}^{N} p_n^{max} - \beta$ on one slot, we have:
$$Q(t'+1) \leq \frac{Vc^{max}\overline{B}^{max}}{p^{min}} + \sum_{n=1}^{N} p_n^{max} - \beta$$
2) $\frac{Vc^{max}\overline{B}^{max}}{p^{min}} < Q(t') \leq \frac{Vc^{max}\overline{B}^{max}}{p^{min}} + \sum_{n=1}^{N} p_n^{max} - \beta$. In this case, since $\phi_n(\alpha_n(t')) \leq 1$, there is no possibility that $Vc_n\overline{B}_n\phi_n(\alpha_n(t')) \geq Q(t')p_n(\alpha_n(t'))$ unless $\alpha_n(t') = 0$. Thus, the Lyapunov indexing algorithm of minimizing (29) chooses $\alpha_n(t') = 0$ for all n. Thus, all indices are 0. This implies that $Q(t'+1)$ cannot increase, and we get $Q(t'+1) \leq \frac{Vc^{max}\overline{B}^{max}}{p^{min}} + \sum_{n=1}^{N} p_n^{max} - \beta$.

■

Theorem 2: The proposed Lyapunov indexing algorithm achieves the constraint:

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}p_n(\alpha_n(t))\leq\beta$$

Proof: Using Lemma 1 under the special case that each frame only occupies one slot, we get that if $\{Q(t)\}_{t=0}^{\infty}$ is deterministically bounded, then the time average constraint is satisfied. Then, according to Lemma 3 we are done. ■

IV. Multi-user optimality in a special case

In general, it is very difficult to prove optimality of the above multi-user algorithm. There are mainly two reasons. The first reason is that multiple users might renew themselves asynchronously, making it difficult to define a "renewal frame" for the whole system. Thus, the proof technique in Theorem 1 is infeasible. The second reason is that, even without the time average constraint, the problem degenerates into a standard restless bandit problem where the optimality of indexing is not guaranteed.

This section considers a special case of the multi-user file downloading problem where the Lyapunov indexing algorithm is provably optimal. The special case has no time average power constraint. Further, for each user $n\in\{1,\ldots,N\}$:

- Each file consists of a random number of fixed length packets with mean $\overline{B}_n=1/\mu_n$.
- The decision set $\mathcal{A}_n=\{0,1\}$, where 0 stands for "idle" and 1 stands for "download." If $\alpha_n(t)=1$, then user n successfully downloads a single packet.
- $\phi_n(\alpha_n(t))=\mu_n\alpha_n(t)$.
- Idle time is geometrically distributed with mean $1/\lambda_n$.
- The special case $\mu_n=1-\lambda_n$ *is assumed.*

The assumption that the file length and idle time parameters μ_n and λ_n satisfy $\mu_n=1-\lambda_n$ is restrictive. However, when this assumption holds, there exists a certain queueing system which admits *exactly the same Markov dynamics* as the system considered here (described in Section IV-A below). More importantly, it allows us to implement the stochastic coupling idea to prove optimality.

The goal is to maximize the sum throughput (in units of packets/slot), which is defined as:

$$\liminf_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\overline{B}_n\phi(\alpha_n(t)). \tag{31}$$

In this special case, the multi-user file downloading problem reduces to the following:

$$\text{Max: }\liminf_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\alpha_n(t) \tag{32}$$

$$\text{S.t.: }\sum_{n=1}^{N}\alpha_n(t)\leq M\quad\forall t\in\{0,1,2,\cdots\} \tag{33}$$

$$\alpha_n(t)\in\{0,F_n(t)\} \tag{34}$$

$$Pr[F_n(t+1)=1\mid F_n(t)=0]=\lambda_n \tag{35}$$

$$Pr[F_n(t+1)=0\mid F_n(t)=1]=\alpha_n(t)(1-\lambda_n) \tag{36}$$

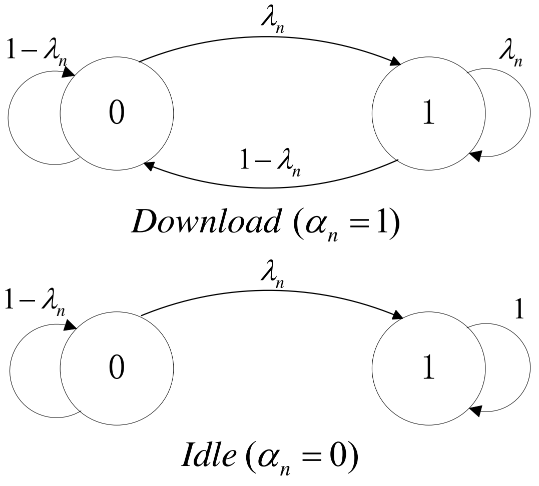

Fig. 2. Markovian dynamics of the n-th system.

where the equality (36) uses the fact that $\mu_n=1-\lambda_n$. A picture that illustrates the Markov structure of constraints (34)-(36) is given in Fig. 2

A. A system with N single-buffer queues

The above model, with the assumption $\mu_n=1-\lambda_n$, is structurally equivalent to the following: Consider a system of N single-buffer queues, M servers, and independent Bernoulli packet arrivals with rates λ_n to each queue $n\in\{1,\ldots,N\}$. This considers *packet arrivals* rather than *file arrivals*, so there are no file length variables and no parameters μ_n in this interpretation. Let $\mathbf{A}(t)=(A_1(t),\ldots,A_N(t))$ be the binary-valued vector of packet arrivals on slot t, assumed to be i.i.d. over slots and independent in each coordinate. Assume all packets have the same size and each queue has a single buffer that can store just one packet. Let $F_n(t)$ be 1 if queue n has a packet at the beginning of slot t, and 0 else. Each server can transmit at most 1 packet per slot. Let $\alpha_n(t)$ be 1 if queue n is served on slot t, and 0 else. An arrival $A_n(t)$ occurs at the end of slot t and is accepted only if queue n is empty at the end of the slot (such as when it was served on that slot). Packets that are not accepted are dropped. The Markov dynamics are described by the same figure as before, namely, Fig. 2. Further, the problem of maximizing throughput is given by the *same* equations (32)-(36). Thus, although the variables of the two problems have different interpretations, the problems are structurally equivalent. For simplicity of exposition, the remainder of this section uses this single-buffer queue interpretation.

B. Optimality of the indexing algorithm

Since there is no power constraint, for any $V>0$ the Lyapunov indexing policy (30) in Section III-A reduces to the following (using $c_n=1$, $Q(t)\equiv0$): If there are fewer than M non-empty queues, serve all of them. Else, serve the M non-empty queues with the largest values of γ_n, where:

$$\gamma_n=\frac{1}{1+(1-\lambda_n)/\lambda_n}=\lambda_n.$$

Thus, the Lyapunov indexing algorithm in this context reduces to serving the (at most M) non-empty queues with the largest λ_n values each time slot. For the remainder of this section, this is called the Max-λ policy. The following theorem shows that Max-λ is optimal in this context.

Theorem 3: The Max-λ policy is optimal for the problem (32)-(36). In particular, under the single-buffer queue interpretation, it maximizes throughput over all policies that transmit on each slot t without knowledge of the arrival vector $\mathbf{A}(t)$.

For the N single-buffer queue interpretation, the total throughput is equal to the raw arrival rate $\sum_{i=1}^{N} \lambda_i$ minus the packet drop rate. Intuitively, the reason Max-λ is optimal is that it chooses to leave packets in the queues that are least likely to induce packet drops. An example comparison of the throughput gap between Max-λ and Min-λ policies is given in Appendix A.

The proof of Theorem 3 is divided into two parts. The first part uses stochastic coupling techniques to prove that Max-λ dominates all alternative *work-conserving* policies. A policy is *work-conserving* if it does not allow any server to be idle when it could be used to serve a non-empty queue. The second part of the proof shows that throughput cannot be increased by considering non-work-conserving policies.

C. Preliminaries on stochastic coupling

Consider two discrete time processes $\mathcal{X} \triangleq \{X(t)\}_{t=0}^{\infty}$ and $\mathcal{Y} \triangleq \{Y(t)\}_{t=0}^{\infty}$. The notation $\mathcal{X} =_{st} \mathcal{Y}$ means that $\mathcal{X}$ and $\mathcal{Y}$ are *stochastically equivalent*, in that they are described by the same probability law. Formally, this means that their joint distributions are the same, so for all $t \in \{0, 1, 2, \ldots\}$ and all $(z_0, \ldots, z_t) \in \mathcal{R}^{t+1}$:

$$\begin{aligned} &Pr[X(0) \leq z_0, \ldots, X(t) \leq z_t] \\ &= Pr[Y(0) \leq z_0, \ldots, Y(t) \leq z_t] \end{aligned}$$

The notation $\mathcal{X} \leq_{st} \mathcal{Y}$ means that $\mathcal{X}$ is *stochastically less than or equal to* $\mathcal{Y}$, as defined by the following theorem.

Theorem 4: ([4]) The following three statements are equivalent:

1) $\mathcal{X} \leq_{st} \mathcal{Y}$.
2) $Pr[g(X(0), X(1), \cdots, X(t)) > z] \leq Pr[g(Y(0), Y(1), \cdots, Y(t)) > z]$ for all $t \in \mathbb{Z}^+$, all z, and for all functions $g : \mathcal{R}^n \to \mathcal{R}$ that are measurable and nondecreasing in all coordinates.
3) There exist two stochastic processes $\mathcal{X}'$ and $\mathcal{Y}'$ on a common probability space that satisfy $\mathcal{X} =_{st} \mathcal{X}'$, $\mathcal{Y} =_{st} \mathcal{Y}'$, and $X'(t) \leq Y'(t)$ for every $t \in \mathbb{Z}^+$.

The following additional notation is used in the proof of Theorem 3.

- Arrival vector $\{\mathbf{A}(t)\}_{t=0}^{\infty}$, where $\mathbf{A}(t) \triangleq [A_1(t)\ A_2(t)\ \cdots\ A_N(t)]$. Each $A_n(t)$ is an independent binary random variable that takes 1 w.p. λ_n and 0 w.p. $1 - \lambda_n$.
- Buffer state vector $\{\mathbf{F}(t)\}_{t=0}^{\infty}$, where $\mathbf{F}(t) \triangleq [F_1(t)\ F_2(t)\ \cdots\ F_N(t)]$. So $F_n(t) = 1$ if queue n has a packet at the beginning of slot t, and $F_n(t) = 0$ else.
- Total packet process $\mathcal{U} \triangleq \{U(t)\}_{t=0}^{\infty}$, where $U(t) \triangleq \sum_{n=1}^{N} F_n(t)$ represents the total number of packets in the system on slot t. Since each queue can hold at most one packet, we have $0 \leq U(t) \leq N$ for all slots t.

D. Stochastic ordering of buffer state process

The next lemma is the key to proving Theorem 3. The lemma considers the multi-queue system with a fixed but arbitrary initial buffer state $\mathbf{F}(0)$. The arrival process $\mathbf{A}(t)$ is as defined above. Let $\mathcal{U}^{\text{Max-}\lambda}$ be the total packet process under the Max-λ policy. Let $\mathcal{U}^{\pi}$ be the corresponding process starting from the same initial state $\mathbf{F}(0)$ and having the same arrivals $\mathbf{A}(t)$, but with an arbitrary work-conserving policy π.

Lemma 4: The total packet processes $\mathcal{U}^{\pi}$ and $\mathcal{U}^{\text{Max-}\lambda}$ satisfy:

$$\mathcal{U}^{\pi} \leq_{st} \mathcal{U}^{\text{Max-}\lambda} \tag{37}$$

Proof: Without loss of generality, assume the queues are sorted so that $\lambda_n \leq \lambda_{n+1}$, $n = 1, 2, \cdots, N-1$. Define $\{\mathbf{F}^{\pi}(t)\}_{t=0}^{\infty}$ as the buffer state vector under policy π. Define $\{\mathbf{F}^{\text{Max-}\lambda}(t)\}_{t=0}^{\infty}$ as the corresponding buffer states under the Max-λ policy. By assumption the initial states satisfy $\mathbf{F}^{\pi}(0) = \mathbf{F}^{\text{Max-}\lambda}(0)$. Next, we construct a *third process* $\mathcal{U}^{\lambda}$ with a *modified* arrival vector process $\{\mathbf{A}^{\lambda}(t)\}_{t=0}^{\infty}$ and a corresponding buffer state vector $\{\mathbf{F}^{\lambda}(t)\}_{t=0}^{\infty}$ (with the same initial state $\mathbf{F}^{\lambda}(0) = \mathbf{F}^{\pi}(0)$), which satisfies:

1) $\mathcal{U}^{\lambda}$ is also generated from the Max-λ policy.
2) $\mathcal{U}^{\lambda} =_{st} \mathcal{U}^{\text{Max-}\lambda}$. Since the total packet process is completely determined by the initial state, the scheduling policy, and the arrival process, it suffices to show that $\{\mathbf{A}^{\lambda}(t)\}_{t=0}^{\infty}$ and $\{\mathbf{A}(t)\}_{t=0}^{\infty}$ have the same probability law.
3) $U^{\pi}(t) \leq U^{\lambda}(t)\ \forall t \geq 0$.

Since the arrival process $\mathbf{A}(t)$ is i.i.d. over slots, in order to guarantee 2) and 3), it is sufficient to construct $\mathbf{A}^{\lambda}(t)$ coupled with $\mathbf{A}(t)$ for each t so that the following two properties hold for all $t \geq 0$:

- The random variables $\mathbf{A}(t)$ and $\mathbf{A}^{\lambda}(t)$ have the same probability law. Specifically, both produce arrivals according to Bernoulli processes that are independent over queues and over time, with $Pr[A_n(t) = 1] = Pr[A_n^{\lambda}(t) = 1] = \lambda_n$ for all $n \in \{1, \ldots, N\}$.
- For all $j \in \{1, 2, \cdots, N\}$,

$$\sum_{n=1}^{j} F_n^{\pi}(t) \leq \sum_{n=1}^{j} F_n^{\lambda}(t), \tag{38}$$

The construction is based on an induction.

At $t = 0$ we have $\mathbf{F}^{\pi}(0) = \mathbf{F}^{\lambda}(0)$. Thus, (38) naturally holds for $t = 0$. Now fix $\tau \geq 0$ and assume (38) holds for all slots up to time $t = \tau$. If $\tau \geq 1$, further assume the arrivals $\{\mathbf{A}^{\lambda}(t)\}_{t=0}^{\tau-1}$ have been constructed to have the same probability law as $\{\mathbf{A}(t)\}_{\tau=0}^{\tau-1}$. Since arrivals on slot τ occur at the *end* of slot τ, the arrivals $\mathbf{A}^{\lambda}(\tau)$ must be constructed. We are going to show there exists an $\mathbf{A}^{\lambda}(\tau)$ that is *coupled* with $\mathbf{A}(\tau)$ so that it has the same probability law and it also ensures (38) holds for $t = \tau + 1$.

Since arrivals occur after the transmitting action, we divide the analysis into two parts. First, we analyze the temporary buffer states after the transmitting action *but before arrivals* *occur*. Then, we define arrivals $\mathbf{A}^{\lambda}(\tau)$ at the end of slot τ to achieve the desired coupling.

Define $\tilde{\mathbf{F}}^{\pi}(\tau)$ and $\tilde{\mathbf{F}}^{\lambda}(\tau)$ as the *temporary buffer states* right after the transmitting action at slot τ but before arrivals occur under policy π and policy Max-λ, respectively. Thus, for each queue $n \in \{1, \ldots, N\}$:

$$\tilde{F}_n^{\pi}(\tau) = F_n^{\pi}(\tau) - \alpha_n^{\pi}(\tau) \quad (39)$$
$$\tilde{F}_n^{\lambda}(\tau) = F_n^{\lambda}(\tau) - \alpha_n^{\lambda}(\tau) \quad (40)$$

where $\alpha_n^{\pi}(\tau)$ and $\alpha_n^{\lambda}(\tau)$ are the slot τ decisions under policy π and Max-λ, respectively. Since (38) holds for $j = N$ on slot τ, the total number of packets at the start of slot τ under policy π is less than or equal to that under Max-λ. Since both policies π and Max-λ are work-conserving, it is impossible for policy π to transmit more packets than Max-λ during slot τ. This implies:

$$\sum_{n=1}^{N} \tilde{F}_n^{\pi}(\tau) \leq \sum_{n=1}^{N} \tilde{F}_n^{\lambda}(\tau). \quad (41)$$

Indeed, if π transmits the same number of packets as Max-λ on slot τ, then (41) clearly holds. On the other hand, if π transmits *fewer* packets than Max-λ, it must transmit fewer than M packets (since M is the number of servers). In this case, the work-conserving nature of π implies that all non-empty queues were served, so that $\tilde{F}_n^{\pi}(\tau) = 0$ for all n and (41) again holds. We now claim the following holds:

Lemma 5:

$$\sum_{n=1}^{j} \tilde{F}_n^{\pi}(\tau) \leq \sum_{n=1}^{j} \tilde{F}_n^{\lambda}(\tau) \quad \forall j \in \{1, 2, \cdots, N\}. \quad (42)$$

Proof: See Appendix B. ■

Now let $j^{\pi}(l)$ and $j^{\lambda}(l)$ be the subscript of l-th *empty* temporary buffer (with order starting from the first queue) corresponding to $\tilde{\mathbf{F}}^{\pi}(\tau)$ and $\tilde{\mathbf{F}}^{\lambda}(\tau)$, respectively. It follows from (42) that the π system on slot τ has at least as many empty temporary buffer states as the Max-λ policy, and:

$$j^{\pi}(l) \leq j^{\lambda}(l) \quad \forall l \in \{1, 2, \cdots, K(\tau)\} \quad (43)$$

where $K(\tau) \leq N$ is the the number of empty temporary buffer states under Max-λ at time slot τ. Since $\lambda_i \leq \lambda_j$ if and only if $i \leq j$, (43) further implies that

$$\lambda_{j^{\pi}(l)} \leq \lambda_{j^{\lambda}(l)} \quad \forall l \in \{1, 2, \cdots, K(\tau)\}. \quad (44)$$

Now construct the arrival vector $\mathbf{A}^{\lambda}(\tau)$ for the system with the Max-λ policy in the following way:

$$A_{j^{\pi}(l)}(\tau) = 1 \Rightarrow A^{\lambda}_{j^{\lambda}(l)}(\tau) = 1 \quad w.p.\ 1 \quad (45)$$

$$A_{j^{\pi}(l)}(\tau) = 0 \Rightarrow \begin{cases} A^{\lambda}_{j^{\lambda}(l)}(\tau) = 0, & \text{w.p. } \frac{1-\lambda_{j^{\lambda}(l)}}{1-\lambda_{j^{\pi}(l)}}; \\ A^{\lambda}_{j^{\lambda}(l)}(\tau) = 1, & \text{w.p. } \frac{\lambda_{j^{\lambda}(l)}-\lambda_{j^{\pi}(l)}}{1-\lambda_{j^{\pi}(l)}}. \end{cases} \quad (46)$$

Notice that (46) uses valid probability distributions because of (44). This establishes the slot τ arrivals for the Max-λ policy for all of its $K(\tau)$ queues with empty temporary buffer states. The slot τ arrivals for its queues with non-empty temporary buffers will be dropped and hence do not affect the queue states on slot $\tau + 1$. Thus, we define arrivals $A_j^{\lambda}(\tau)$ to be independent of all other quantities and to be Bernoulli with $Pr[A_j^{\lambda}(\tau) = 1] = \lambda_j$ for all j in the set:

$$j \in \{1, 2, \cdots, N\} \setminus \{j^{\lambda}(1), \cdots, j^{\lambda}(K(\tau))\}$$

Now we verify that $\mathbf{A}(\tau)$ and $\mathbf{A}^{\lambda}(\tau)$ have the same probability law. First condition on knowledge of $K(\tau)$ and the particular $j^{\pi}(l)$ and $j^{\lambda}(l)$ values for $l \in \{1, \ldots, K(\tau)\}$. All queues j with non-empty temporary buffer states on slot τ under Max-λ were defined to have arrivals $A_j^{\lambda}(\tau)$ as independent Bernoulli variables with $Pr[A_j^{\lambda}(\tau) = 1] = \lambda_j$. It remains to verify those queues within $\{j^{\lambda}(1), \cdots, j^{\lambda}(K(\tau))\}$. According to (46), for any queue $j^{\lambda}(l)$ in set $\{j^{\lambda}(1), \cdots, j^{\lambda}(K(\tau))\}$, it follows

$$\begin{aligned} Pr\left[A^{\lambda}_{j^{\lambda}(l)}(\tau) = 0\right] &= (1 - \lambda_{j^{\pi}(l)}) \frac{1 - \lambda_{j^{\lambda}(l)}}{1 - \lambda_{j^{\pi}(l)}} \\ &= 1 - \lambda_{j^{\lambda}(l)} \end{aligned}$$

and so $Pr[A_j^{\lambda}(\tau) = 1] = \lambda_j$ for all $j \in \{j^{\lambda}(l)\}_{l=1}^{K(\tau)}$. Further, mutual independence of $\{A_{j^{\pi}(l)}(\tau)\}_{l=1}^{K(\tau)}$ implies mutual independence of $\{A_{j^{\lambda}(l)}(\tau)\}_{l=1}^{K(\tau)}$. Finally, these quantities are conditionally independent of events before slot τ, given knowledge of $K(\tau)$ and the particular $j^{\pi}(l)$ and $j^{\lambda}(l)$ values for $l \in \{1, \ldots, K(\tau)\}$. Thus, conditioned on this knowledge, $\mathbf{A}(\tau)$ and $\mathbf{A}^{\lambda}(\tau)$ have the same probability law. This holds for all possible values of the conditional knowledge $K(\tau)$ and $j^{\pi}(l)$ and $j^{\lambda}(l)$. It follows that $\mathbf{A}(\tau)$ and $\mathbf{A}^{\lambda}(\tau)$ have the same (unconditioned) probability law.

Finally, we show that the coupling relations (45) and (46) produce such $\mathbf{F}^{\lambda}(\tau + 1)$ satisfying

$$\sum_{n=1}^{j} F_n^{\pi}(\tau+1) \leq \sum_{n=1}^{j} F_n^{\lambda}(\tau+1), \ \forall\, j \in \{1, 2, \cdots, N\}. \quad (47)$$

According to (45) and (46),

$$A_{j^{\pi}(l)}(\tau) \leq A^{\lambda}_{j^{\lambda}(l)}(\tau), \quad \forall l \in \{1, \cdots, K(\tau)\},$$

thus,

$$\sum_{i=1}^{l} A_{j^{\pi}(i)}(\tau) \leq \sum_{i=1}^{l} A^{\lambda}_{j^{\lambda}(i)}(\tau), \quad \forall l \in \{1, \cdots, K(\tau)\}. \quad (48)$$

Pick any $j \in \{1, 2, \cdots, N\}$. Let l^{π} be the number of empty temporary buffers within the first j queues under policy π, i.e.

$$l^{\pi} = \max_{j^{\pi}(l) \leq j} l$$

Similarly define:

$$l^{\lambda} = \max_{j^{\lambda}(l) \leq j} l.$$

Then, it follows:

$$\sum_{n=1}^{j} F_n^{\pi}(\tau+1) = \sum_{n=1}^{j} \tilde{F}_n^{\pi}(\tau) + \sum_{i=1}^{l^{\pi}} A_{j^{\pi}(i)}(\tau) \quad (49)$$

$$\sum_{n=1}^{j} F_n^{\lambda}(\tau+1) = \sum_{n=1}^{j} \tilde{F}_n^{\lambda}(\tau) + \sum_{i=1}^{l^{\lambda}} A^{\lambda}_{j^{\lambda}(i)}(\tau) \quad (50)$$

We know that $l^{\pi} \geq l^{\lambda}$. So there are two cases:

- If $l^\pi = l^\lambda$, then from (49):

$$\begin{aligned}\sum_{n=1}^{j} F_n^\pi(\tau+1) &= \sum_{n=1}^{j} \tilde{F}_n^\pi(\tau) + \sum_{i=1}^{l^\lambda} A_{j^\pi(i)}(\tau) \\ &\leq \sum_{n=1}^{j} \tilde{F}_n^\lambda(\tau) + \sum_{i=1}^{l^\lambda} A_{j^\lambda(i)}(\tau) \\ &= \sum_{n=1}^{j} F_n^\lambda(\tau+1)\end{aligned}$$

where the inequality follows from (42) and from (48) with $l = l^\lambda$. Thus, (47) holds.

- If $l^\pi > l^\lambda$, then from (49):

$$\begin{aligned}\sum_{n=1}^{j} F_n^\pi(\tau+1) &= \sum_{n=1}^{j} \tilde{F}_n^\pi(\tau) + \sum_{i=1}^{l^\lambda} A_{j^\pi(i)}(\tau) \\ &\quad + \sum_{i=l^\lambda+1}^{l^\pi} A_{j^\pi(i)}(\tau) \\ &\leq \sum_{n=1}^{j} \tilde{F}_n^\lambda(\tau) + \sum_{i=1}^{l^\lambda} A_{j^\pi(i)}(\tau) \\ &\leq \sum_{n=1}^{j} \tilde{F}_n^\lambda(\tau) + \sum_{i=1}^{l^\lambda} A^\lambda_{j^\lambda(i)}(\tau) \\ &= \sum_{n=1}^{j} F_n^\lambda(\tau+1).\end{aligned}$$

where the first inequality follows from the fact that

$$\begin{aligned}\sum_{i=l^\lambda+1}^{l^\pi} A_{j^\pi(i)}(\tau) &\leq l^\pi - l^\lambda \\ &= (j - l^\lambda) - (j - l^\pi) \\ &= \sum_{n=1}^{j} \tilde{F}_n^\lambda(\tau) - \sum_{n=1}^{j} \tilde{F}_n^\pi(\tau),\end{aligned}$$

and the second inequality follows from (48).

Thus, (38) holds for $t = \tau + 1$ and the induction step is done. ■

Corollary 1: The Max-λ policy maximizes throughput within the class of work-conserving policies.

Proof: Let $S^\pi(t)$ be the number of packets transmitted under any work-conserving policy π on slot t, and let $S^{\text{Max-}\lambda}(t)$ be the corresponding process under policy Max-λ. Lemma 4 implies $\mathcal{U}^\pi(t) \leq_{st} \mathcal{U}^{\text{Max-}\lambda}$. Then:

$$\begin{aligned}\mathbb{E}\left[S^\pi(t)\right] &= \mathbb{E}\left[\min[U^\pi(t), M]\right] \\ &\leq \mathbb{E}\left[\min[U^{\text{Max-}\lambda}(t), M]\right] \\ &= \mathbb{E}\left[S^{\text{Max-}\lambda}(t)\right]\end{aligned}$$

where the inequality follows from Theorem 4, with the understanding that $g(U(0), \ldots, U(t)) \triangleq \min[U(t), M]$ is a function that is nondecreasing in all coordinates. ■

TABLE I
PROBLEM PARAMETERS

User	λ_n	μ_n	$\phi_n(1)$	c_n	$p_n(1)$
1	0.0028	0.5380	0.4842	4.7527	3.9504
2	0.4176	0.5453	0.4908	2.0681	3.7391
3	0.0888	0.5044	0.4540	2.8656	3.5753
4	0.3181	0.6103	0.5493	2.4605	2.1828
5	0.4151	0.9839	0.8855	4.5554	3.1982
6	0.2546	0.5975	0.5377	3.9647	3.5290
7	0.1705	0.5517	0.4966	1.5159	2.5226
8	0.2109	0.7597	0.6837	3.6364	2.5376

E. Extending to non-work-conserving policies

Corollary 1 establishes optimality of Max-λ over the class of all work-conserving policies. To complete the proof of Theorem 3, it remains to show that throughput cannot be increased by allowing for non-work-conserving policies. It suffices to show that for any non-work-conserving policy, there exists a work-conserving policy that gets the same or better throughput. The proof is straightforward and we give only a proof sketch for brevity. Consider any non-work-conserving policy π, and let $F_n^\pi(t)$ be its buffer state process on slot t for each queue n. For the same initial buffer state and arrival process, define the work-conserving policy π' as follows: Every slot t, policy π' initially allocates the M servers to exactly the same queues as policy π. However, if some of these queues are empty under policy π', it reallocates those servers to any non-empty queues that are not yet allocated servers (in keeping with the work-conserving property). Let $F_n^{\pi'}(t)$ be the buffer state process for queue n under policy π'. It is not difficult to show that $F_n^\pi(t) \geq F_n^{\pi'}(t)$ for all queues n and all slots t. Therefore, on every slot t, the amount of *blocked arrivals* under policy π is always greater than or equal to that under policy π'. This implies the throughput under policy π is less than or equal to that of policy π'.

V. SIMULATION EXPERIMENTS

In this section, we demonstrate near optimality of the multi-user Lyapunov indexing algorithm by extensive simulations. In the first part, we simulate the case in which the file length distribution is geometric, and show that the suboptimality gap is extremely small. In the second part, we test the robustness of our algorithm for more general scenarios in which the file length distribution is not geometric. For simplicity, it is assumed throughout that all transmissions send a fixed sized packet, all files are an integer number of these packets, and that decisions $\alpha_n(t) \in \mathcal{A}_n$ affect the success probability of the transmission as well as the power expenditure.

A. Lyapunov indexing with geometric file length

In the first simulation we use $N = 8$, $M = 4$ with action set $\mathcal{A}_n = \{0, 1\}\ \forall n$; The settings are generated randomly and specified in Table I, and the constraint $\beta = 5$.

The algorithm is run for 1 million slots in each trial and each point is the average of 100 trials. We compare the performance of our algorithm with the optimal randomized policy. The optimal policy is computed by constructing composite states (i.e. if there are three users where user 1 is at state 0, user 2 is at state 1 and user 3 is at state 1, we view 011 as a composite state), and then reformulating this MDP into a linear program (see [18]) with $\mathbf{5985}$ variables and $\mathbf{258}$ constraints.

In Fig. 3, we show that as our tradeoff parameter V gets larger, the objective value approaches the optimal value and achieves a near optimal performance. Fig. 4 and Fig. 5 show that V also affects the virtual queue size and the constraint gap. As V gets larger, the average virtual queue size becomes larger and the gap becomes smaller. We also plot the upper bound of queue size we derived from Lemma 3 in Fig. 4, demonstrating that the queue is bounded. In order to show that V is indeed a trade-off parameter affecting the convergence time, we plotted Fig. 6. It can be seen from the figure that as V gets larger, the number of time slots needed for the running average to roughly converge to the optimal power expenditure becomes larger.

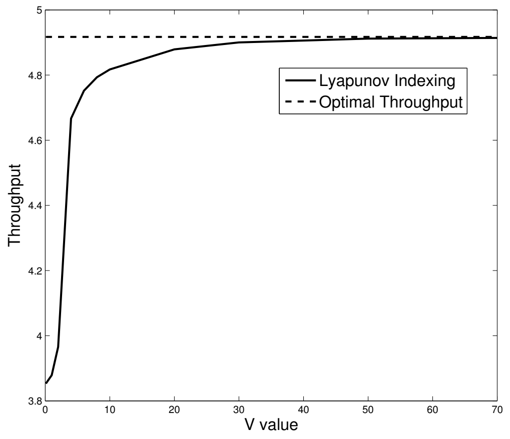

Fig. 3. Throughput versus tradeoff parameter V

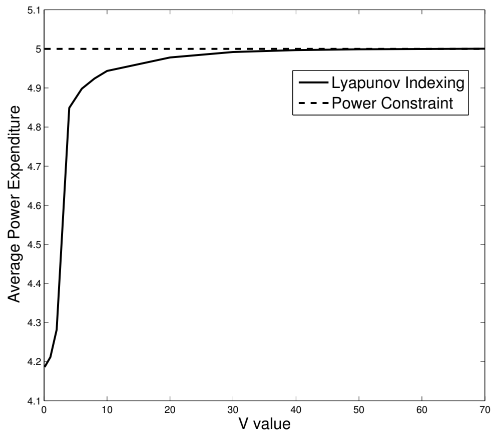

Fig. 4. The time average power consumption versus tradeoff parameter V.

In the second simulation, we explore the parameter space and demonstrate that in general the suboptimality gap of our algorithm is negligible. First, we define the relative error as

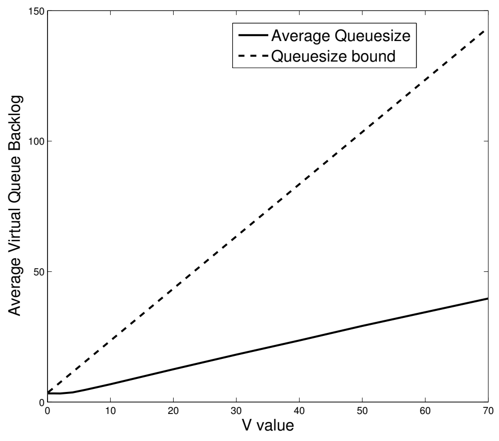

Fig. 5. Average virtual queue backlog versus tradeoff parameter V.

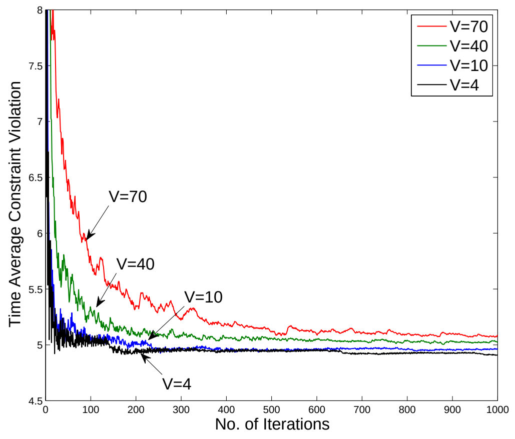

Fig. 6. Running average power consumption versus tradeoff parameter V.

the following:

$$\text{relative error} = \frac{|OBJ - OPT|}{OPT} \tag{51}$$

where OBJ is the objective value after running 1 million slots of our algorithm and OPT is the optimal value. We first explore the system parameters by letting λ_n's and μ_n's take random numbers within 0 and 1, letting c_n take random numbers within 1 and 5, choosing $V = 70$ and fixing the remaining parameters the same as the last experiment. We conduct 1000 Monte-Carlo experiments and calculate the average relative error, which is **0.00083**.

Next, we explore the control parameters by letting the $p_n(1)$ values take random numbers within 2 and 4, letting $\phi_n(1)/\mu_n$ take random numbers between 0 and 1, choosing $V = 70$, and fixing the remaining parameters the same as the first simulation. The relative error is **0.00057**. Both experiments show that the suboptimality gap is extremely small.

B. Lyapunov indexing with non-memoryless file lengths

In this part, we test the sensitivity of the algorithm to different file length distributions. In particular, the uniform

TABLE II
PROBLEM PARAMETERS UNDER GEOMETRIC, UNIFORM AND POISSON DISTRIBUTION

User	μ_n	Unif. interval	Poiss. mean	λ_n	$\phi_n(1)$	c_n	$p_n(1)$
1	1/3	[1,5]	3	0.4955	0.1832	4.3261	2.8763
2	1/2	[1,3]	2	0.1181	0.4187	1.6827	2.0549
3	1/2	[1,3]	2	0.1298	0.4491	1.9483	2.1469
4	1/7	[1,13]	7	0.4660	0.0984	2.7495	3.4472
5	1/4	[1,7]	4	0.1661	0.1742	1.5535	3.2801
6	1/3	[1,5]	3	0.2124	0.3101	4.3151	3.5648
7	1/2	[1,3]	2	0.5295	0.4980	3.6701	2.4680
8	1/5	[1,9]	5	0.2228	0.1971	4.0185	2.2984
9	1/4	[1,7]	4	0.0332	0.1986	3.0411	2.5747

distribution and the Poisson distribution are implemented respectively, while our algorithm still treats them as a geometric distribution with the same mean. We then compare their throughputs with the geometric case.

We use $N = 9$, $M = 4$ with action set $\mathcal{A}_n = \{0, 1\}\ \forall n$. The settings are specified in Table II with constraint $\beta = 5$. Notice that for geometric and uniform distribution, the file lengths are taken to be integer values. The algorithm is run for 1 million slots in each trial and each point is the average of 100 trials.

While the decisions are made using these values, the affect of these decisions incorporates the actual (non-memoryless) file sizes. Fig. 7 shows the throughput-versus-V relation for the two non-memoryless cases and the memoryless case with matched means. The performance of all three is similar. This illustrates that the indexing algorithm is robust under different file length distributions.

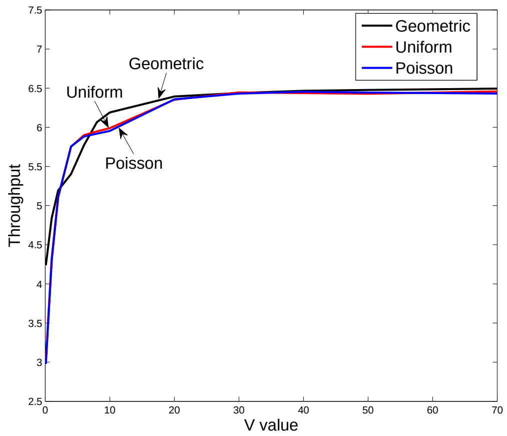

Fig. 7. Throughput versus tradeoff parameter V under different file length distributions.

VI. CONCLUSIONS

We have investigated a file downloading system where the network delays affect the file arrival processes. The single-user case was solved by a variable frame length Lyapunov optimization method. The technique was extended as a well-reasoned heuristic algorithm for the multi-user case. Such heuristics are important because the problem is a multi-dimensional Markov decision problem with very high complexity. The heuristic is simple, can be implemented in an online fashion, and was analytically shown to achieve the desired average power constraint. Moreover, under a special case with no average power constraint, stochastic coupling was used to prove the heuristic is throughput optimal. Simulations suggest that the algorithm is in general very close to optimal. Further, simulations suggest that non-memoryless file lengths can be accurately approximated by the algorithm. These methods can likely be applied in more general situations of restless multi-armed bandit problems with constraints.

APPENDIX A—COMPARISON OF MAX-λ AND MIN-λ

This appendix shows that different work conserving policies can give different throughput for the N single-buffer queue problem of Section IV-A. Suppose we have two single-buffer queues and one server. Let λ_1, λ_2 be the arrival rates of the i.i.d. Bernoulli arrival processes for queues 1 and 2. Assume $\lambda_1 \neq \lambda_2$. There are 4 system states: $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$, where state (i, j) means queue 1 has i packets and queue 2 has j packets. Consider the (work conserving) policy of giving queue 1 strict priority over queue 2. This is equivalent to the Max-λ policy when $\lambda_1 > \lambda_2$, and is equivalent to the Min-λ policy when $\lambda_1 < \lambda_2$. Let $\theta(\lambda_1, \lambda_2)$ be the steady state throughput. Then:

$$\theta(\lambda_1, \lambda_2) = p_{1,0} + p_{0,1} + p_{1,1}$$

where $p_{i,j}$ is the steady state probability of the resulting discrete time Markov chain. One can solve the global balance equations to show that $\theta(1/2, 1/4) > \theta(1/4, 1/2)$, so that the Max-$\lambda$ policy has a higher throughput than the Min-λ policy. In particular, it can be shown that:

- Max-λ throughput: $\theta(1/2, 1/4) = 0.7$
- Min-λ throughput: $\theta(1/4, 1/2) \approx 0.6786$

APPENDIX B—PROOF OF LEMMA 5

This section proves that:

$$\sum_{n=1}^{j} \tilde{F}_n^{\pi}(\tau) \leq \sum_{n=1}^{j} \tilde{F}_n^{\lambda}(\tau) \quad \forall j \in \{1, 2, \cdots, N\}. \tag{52}$$

The case $j = N$ is already established from (41). Fix $j \in \{1, 2, \ldots, N-1\}$. Since π cannot transmit more packets than Max-λ during slot τ, inequality (52) is proved by considering two cases:

1) Policy π transmits less packets than policy Max-λ. Then π transmits less than M packets during slot τ. The work-conserving nature of π implies all non-empty queues were served, so $\tilde{F}_n^{\pi}(\tau) = 0$ for all n and (52) holds.
2) Policy π transmits the same number of packets as policy Max-λ. In this case, consider the temporary buffer states of the last $N - j$ queues under policy Max-λ. If $\sum_{n=j+1}^{N} \tilde{F}_n^{\lambda}(\tau) = 0$, then clearly the following holds

$$\sum_{n=j+1}^{N} \tilde{F}_n^{\pi}(\tau) \geq \sum_{n=j+1}^{N} \tilde{F}_n^{\lambda}(\tau). \tag{53}$$

Subtracting (53) from (41) immediately gives (52). If $\sum_{n=j+1}^{N} \tilde{F}_n^{\lambda}(\tau) > 0$, then all M servers of the Max-λ

system were devoted to serving the largest λ_n queues. So only packets in the last $N - j$ queues could be transmitted by Max-λ during the slot τ. In particular, $\alpha_n^{\lambda}(\tau) = 0$ for all $n \in \{1, \ldots, j\}$, and so (by (40)):

$$\sum_{n=1}^{j} \tilde{F}_n^{\lambda}(\tau) = \sum_{n=1}^{j} F_n^{\lambda}(\tau) \tag{54}$$

Thus:

$$\begin{aligned} \sum_{n=1}^{j} \tilde{F}_n^{\pi}(\tau) &\leq \sum_{n=1}^{j} F_n^{\pi}(\tau) && (55) \\ &\leq \sum_{n=1}^{j} F_n^{\lambda}(\tau) && (56) \\ &= \sum_{n=1}^{j} \tilde{F}_n^{\lambda}(\tau), && (57) \end{aligned}$$

where (55) holds by (39), (56) holds because (38) is true on slot $t = \tau$, and the last equality holds by (54). This proves (52).

References

[1] X. Wei and M. J. Neely. Power aware wireless file downloading: A constrained restless bandit approach. In *Proc. Intl. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Hammamet, Tunisia, May 2014.

[2] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.

[3] M. J. Neely. Dynamic optimization and learning for renewal systems. *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 32-46, Jan. 2013.

[4] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466-478, March 1993.

[5] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, no. 4, pp. 401-457, 2005.

[6] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[7] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, pp. 1333-1344, Dec. 2007.

[8] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396-409, April 2008.

[9] S. Liu, L. Ying, and R. Srikant. Throughput-optimal opportunistic scheduling in the presence of flow-level dynamics. *IEEE/ACM Trans. Networking*, vol. 19, no. 4, pp. 1057-1070, Jan. 2011.

[10] L. Huang and M. J. Neely. Utility optimal scheduling in energy-harvesting networks. *IEEE/ACM Trans. Networking*, vol. 21, no. 4, pp. 117-1130, Aug. 2013.

[11] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2005.

[12] M. J. Neely. Asynchronous control for coupled Markov decision systems. In *Proc. Information Theory Workshop (ITW)*, pages pp. 287–291, Sep. 2012.

[13] D. Dolgov and E. Durfee. Optimal resource allocation and policy formulation in loosely-coupled Markov decision processes. In *Proc. ICAPS*, pages pp. 315–324, June 2004.

[14] N. Meuleau, M. Hauskrecht, K.-E. Kim, L. Peshkin, L. P. Kaelbling, T. Dean, and C. Boutilier. Solving very large weakly coupled Markov decision processes. In *Proc. 15th National Conf. on Artificial Intelligence*, 1998.

[15] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, vol. 25, pp. 287-298, 1988.

[16] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queueing network control. *Math. Oper. Res.*, vol. 24, no. 2, pp. 293-305, May 1999.

[17] M. J. Neely. Stability and probability 1 convergence for queueing networks via Lyapunov optimization. *Journal of Applied Mathematics*, doi:10.1155/2012/831909, 2012.

[18] B. Fox. Markov renewal programming by linear fractional programming. *SIAM Journal on Applied Mathematics*, vol. 14, no. 6, pp. 1418-1432, Nov. 1966.